Self-Organizing Sensor Networks: Efficient Distributed Mechanisms

Alvin S. Lim
Computer Science and Software Engineering
Auburn University
lim@eng.auburn.edu

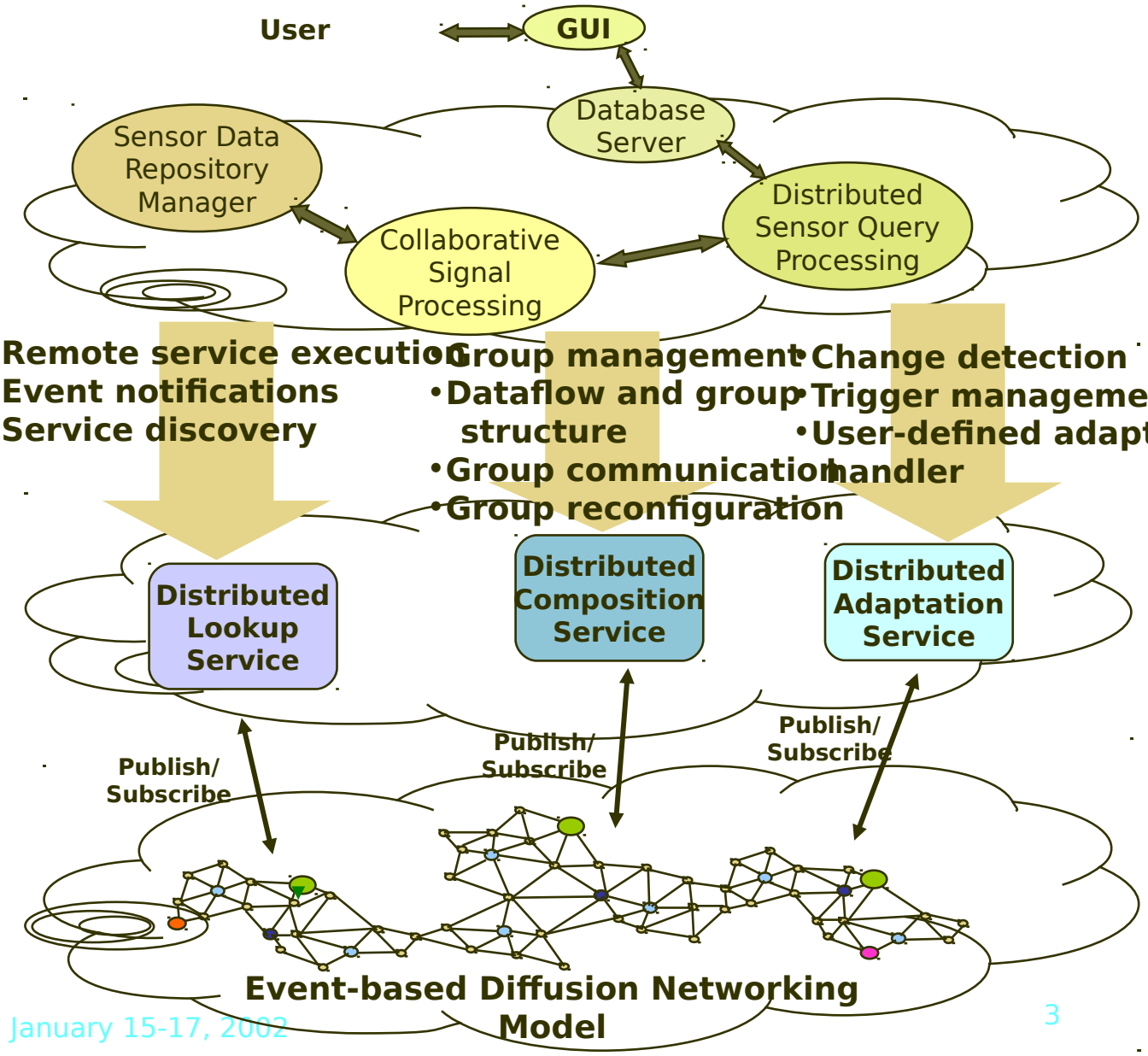
January 16, 2002

DARPA/ITO
Sensor Information Technology
PI Meeting
Santa Fe, NM

Objectives

- **Implement the following distributed services for self-organization on top of a dynamic network routing protocol (directed diffusion)**
  - Interprocess communication: Remote execution (client server interaction), events notification
  - Distributed lookup service (enable spontaneous service creation and discovery)
  - Distributed composition service (simplify group interaction and adaptation in task requirements)
  - Distributed adaptation service (responsive to failures, reconfiguration, mobility, network changes)
- **Provide appropriate abstraction for self-organization of application systems**
- **Improve performance: Reduce overall latency time and network traffic using these distributed services**

Self-Organizing Distributed Services

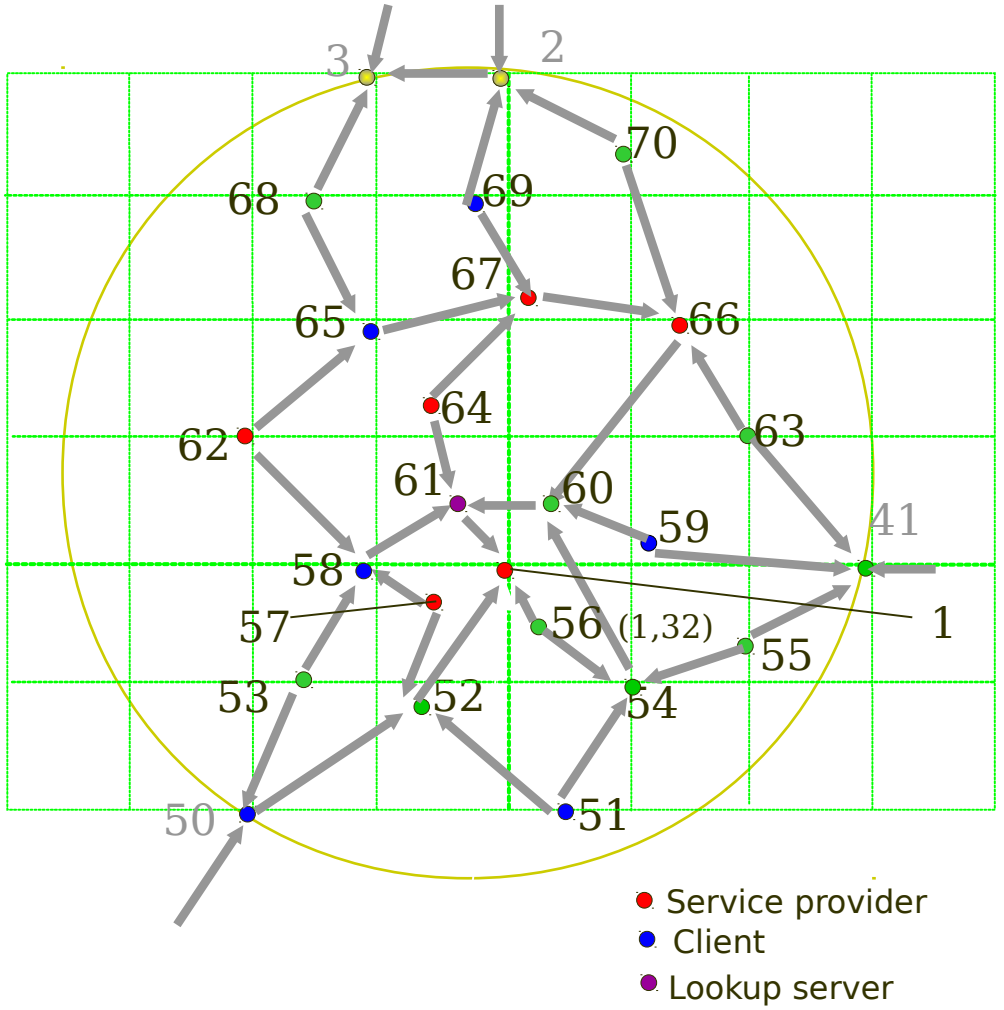


Distributed Lookup Service

- **Software status:**
  - **Development completed -- undergoing software testing**
  - **Future enhancement: other interface type using mobile codes and interface definition languages**
  - **Will be using PSU/ARL mobile code to implement remote execution of distributed service as an alternate and more flexible interface type**
- **Successfully executed UTK/LSU mobile agents with dynamic migration using Auburn lookup service to locate mobile agent demons and to migrate agents (using service_exec() on top of diffusion network)**
  - **Itinerary of mobile agent may be updated on the fly with changes in the dynamic cluster membership**
- **Showed ease of implementing dynamic mobile agents using lookup service and service exec**
- **Will test it with other Sensit software (e.g. mobile/distributed query processing, classification/tracking algorithms) as they are become available**
- **Tested at 29 Palms**

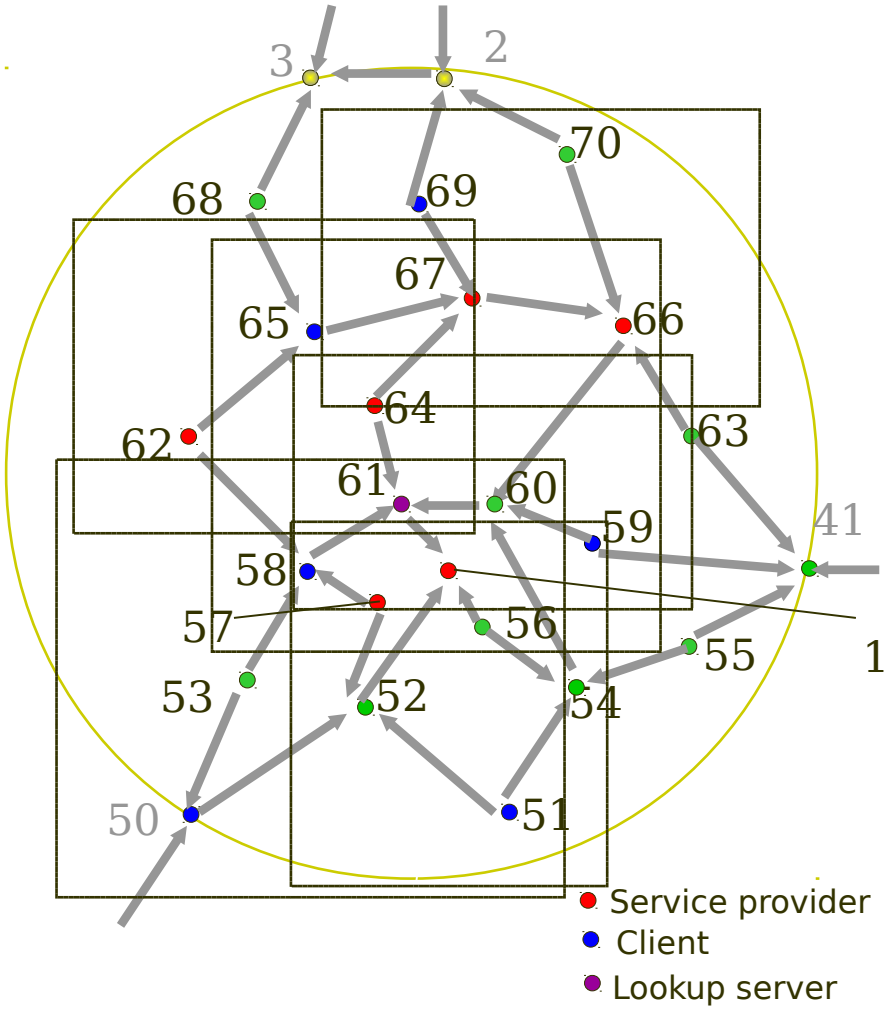
Sitex02 at 29 Palms, November 2001: Experiments on Distributed Lookup Services

- **Demonstrate distributed lookup services and remote service exec over diffusion-based sensor networks and their usefulness for collaborative CSIP algorithms**
- **Multiple concurrent sets of data clients in 30 nodes:**
  - **Locate respective data providers through lookup server**
  - **6 clients continually issue different queries to specific service providers through remote service exec built on top of diffusion network.**



Region Filters

- **Use region filters that process the data clients and providers location info (retrieved from the lookup server) to reduce network traffic by restricting flooding of interests**
- **Location info of providers are cached by the service lookup and service exec functions, transparent to the application**



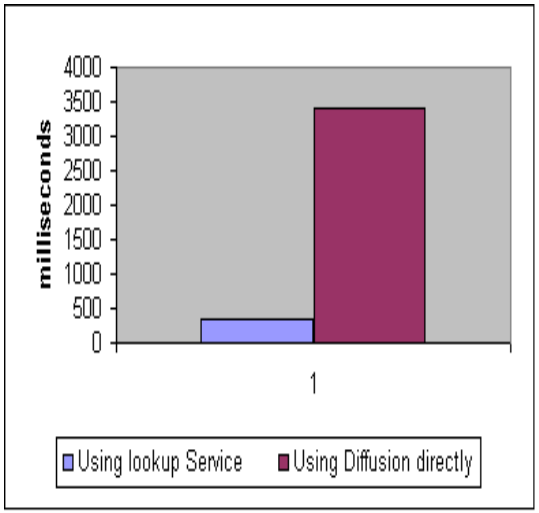
Performance Measurements and Demonstration

- Measure response time, throughput and network traffic while a set of 6 pairs of data clients and providers are continually communicating.
- Two types of application scenarios were compared:
  - using lookup service
  - using diffusion directly
- Demonstrated UTK/LSU mobile agent (executing classification algorithms) migration and itinerary adaptation using lookup service and remote service exec over diffusion networks

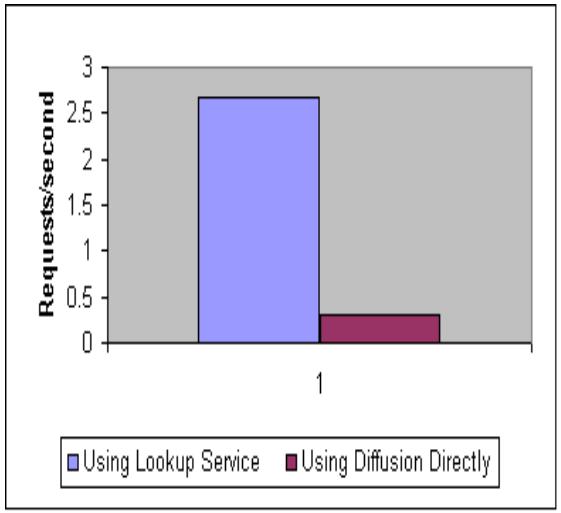
Results from 29 Palms Sitex02 experiments

- The following results show that using lookup service, concurrent CSIP applications can reduce delay, increase throughput and reduce network traffic compared to applications using diffusion directly.
- Demonstrates there are many opportunities for improving the performance of distributed application on diffusion using distributed services (e.g. lookup service, remote service execution, etc.)

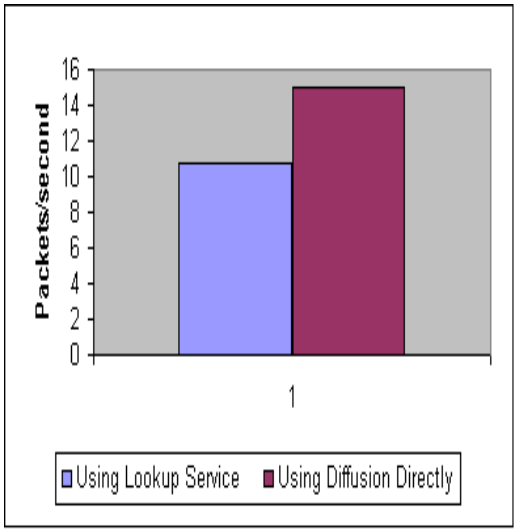
Average Response Delay



Client Average Throughput



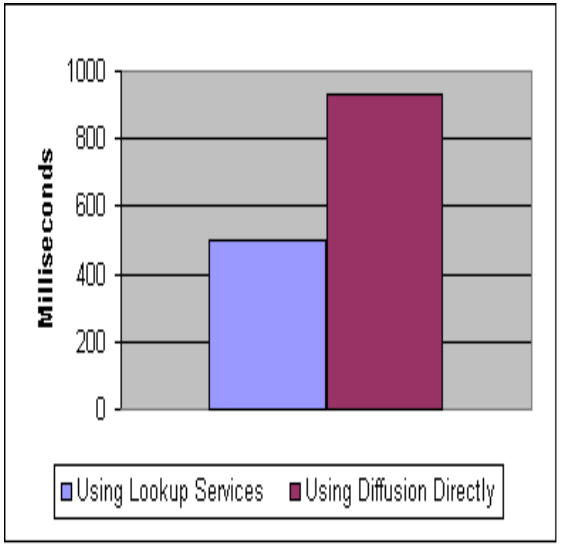
Average Network Traffic



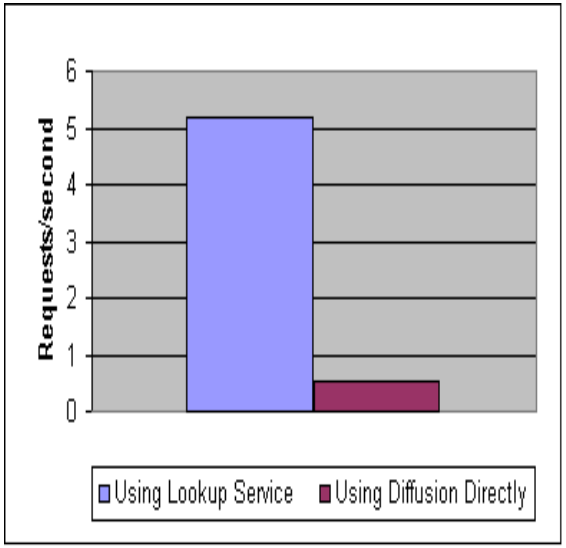
Laboratory Results (1)

- The experiments were repeated in our lab using Linux 400 Mhz computers to simulate the same 29 Palm network configuration
  - Continual query of cache server – 6 pairs of client-server
- The results are similar to those from 29 Palms

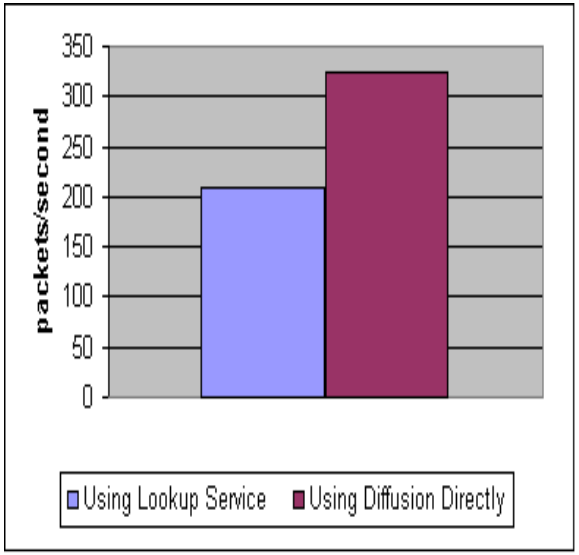
Average Response Delay



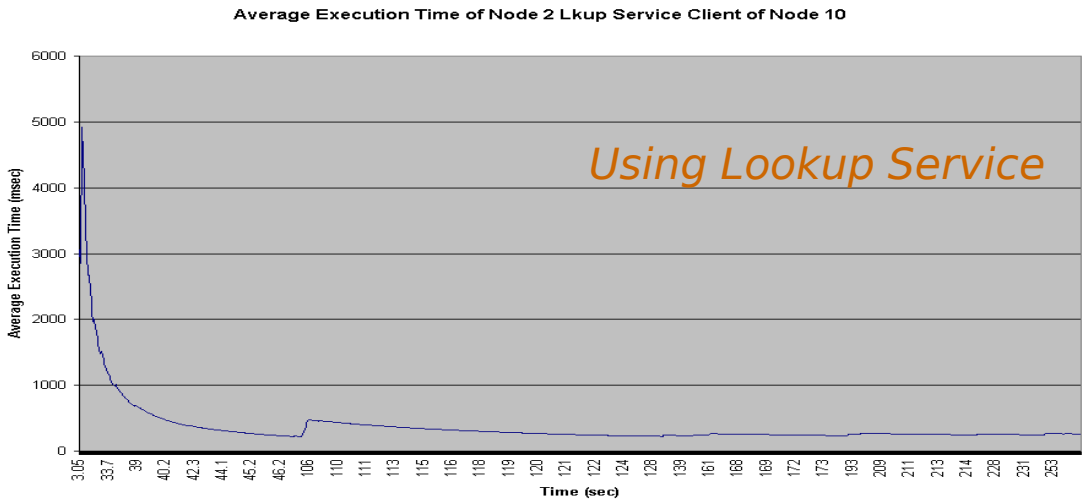
Client Average Throughput

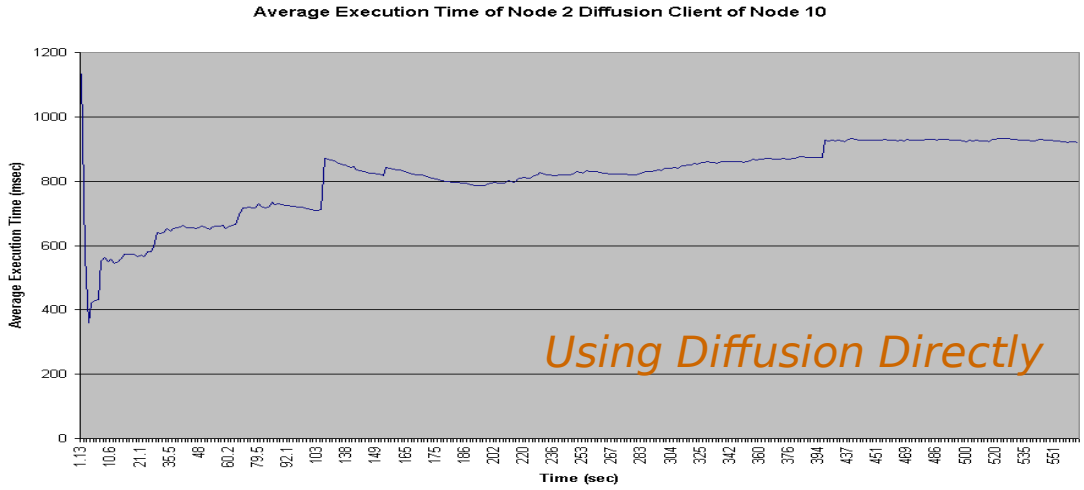


Average Network Traffic

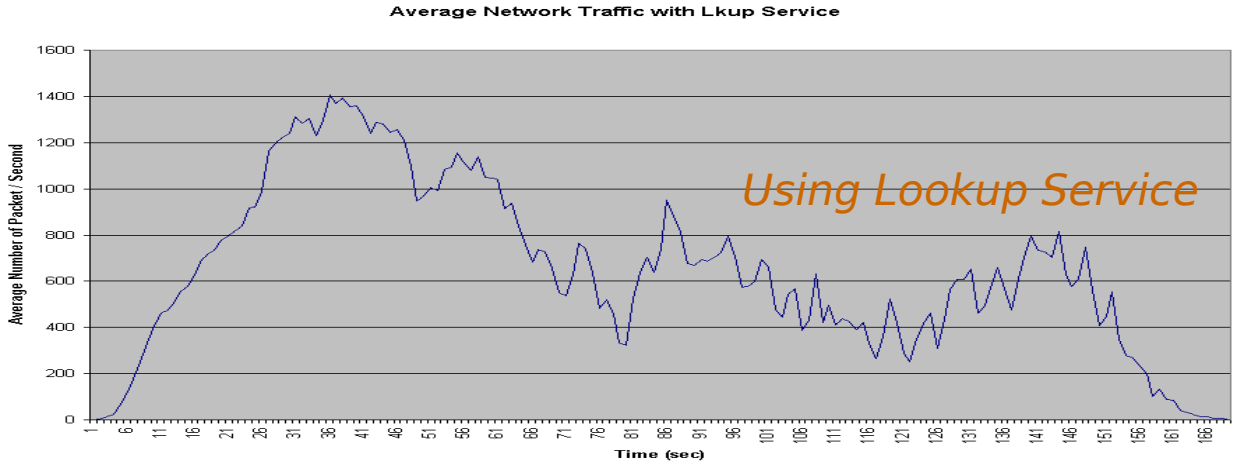


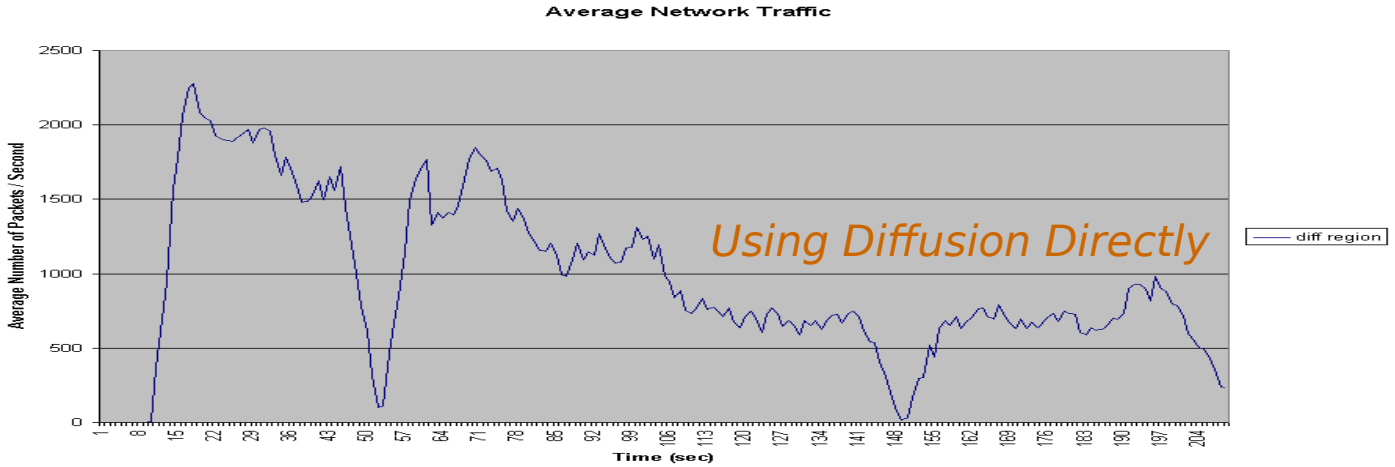
Lab Experiment Results (1) — Average Response Delay (msec)





b Experiment Results (2) — Average Network Traffic (packets/se

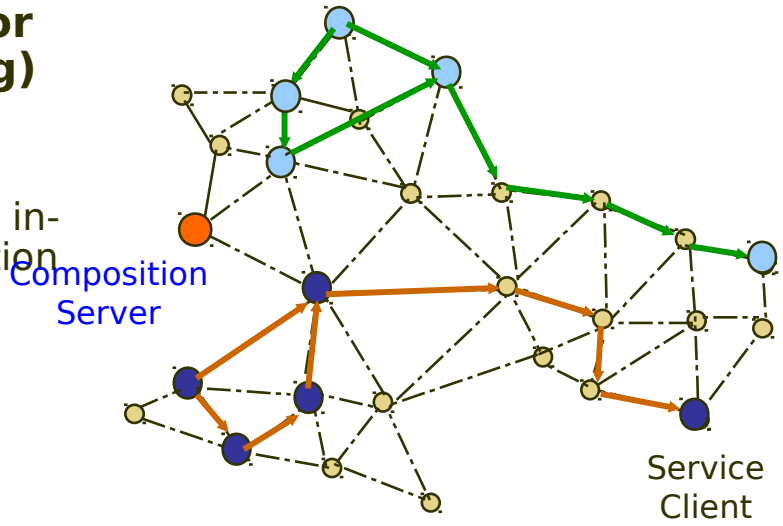




- Show that appropriate distributed mechanisms can improve performance
- Advantage for implementing these mechanisms in the distributed services

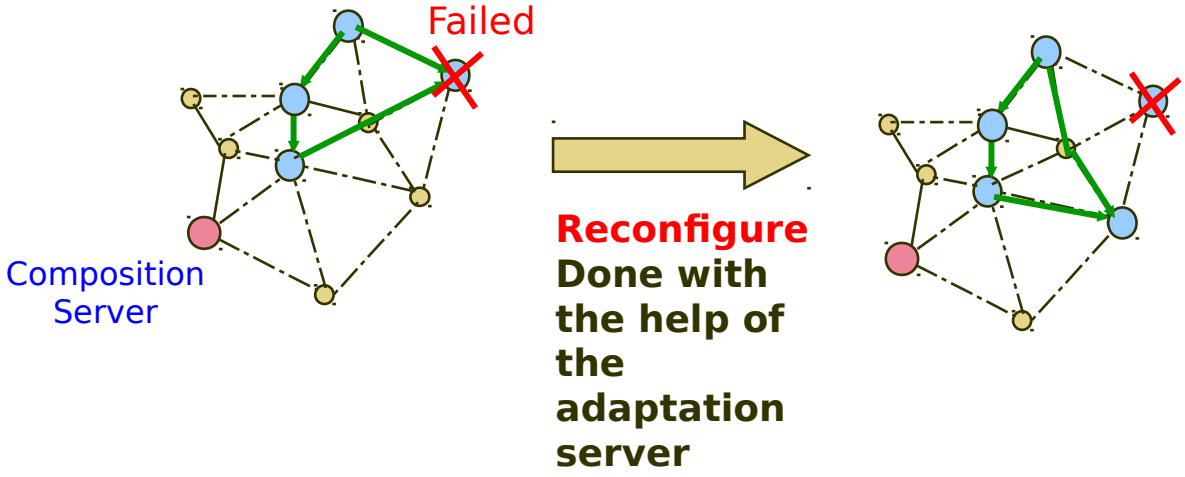
Composition Services

- **Create groups dynamically (e.g. for local collaboration, global tracking) with composition server**
  - Dynamically join/leave group
  - Group communication: exploit diffusion in-network processing and group information
- **Data flow structures within group (e.g. for continuous monitoring)**
  - Create and maintain streams between tasks
  - Dependencies between tasks
  - Primitives for composition of data flow streams, e.g. split, merge, filter, buffer



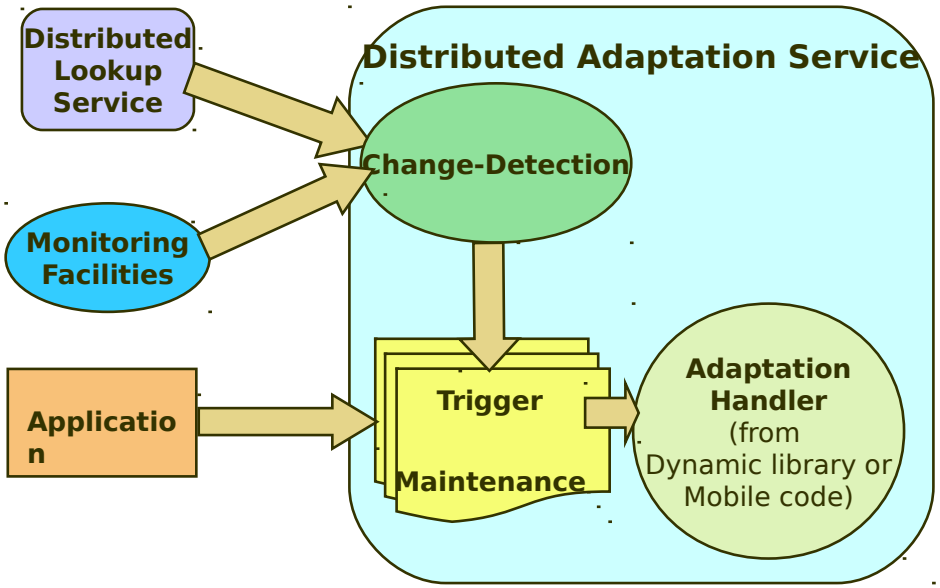
Dynamic Reconfiguration of groups and composition structure

- **Group composition structures maintained by the composition server useful for dynamic reconfiguration**
  - Continuous operations with automatic recovery of consistency (application-specific semantics)



Adaptation Services

- Application registers a condition trigger and adaptation handler with the adaptation server
- Change detection requires event inputs from the distributed lookup servers or monitoring facilities, e.g. density, SNR, fault rate
- A matched condition will trigger the respective adaptation handler which will notify the affected nodes to execute the adaptation operations, e.g. change in task algorithms



Support for Adaptation in Collaborative Signal Processing

- **Sensors may fail, incrementally deployed or dynamically reconfigured**
- **Dynamic steering: Distributed sensor applications steer around changes in the sensor network, such as mobility, failure, density, certainty, and reconfiguration**
- **Dynamic clustering: Active re-clustering of sensors based on density and level of activities to reduce collaborative processing and communication costs**
- **Dynamic tasking: Implement changes in task requirements of fielded sensors by dynamically downloading and executing codes to targeted sensors (PSU/ARL)**

Support for Dynamic Multi-Sensor Fusion

- **Several algorithms available for multi-sensor fusion, e.g. Dolev's algorithms, fast convergence algorithm, optimal sensor fusion algorithm, Brooks-Iyengar hybrid and Dempster-Shafer algorithm**
- **Choice of algorithm depends on: Availability, density, and uncertainty of the sensors**
  - **May be cached and retrieved from the distributed lookup servers**
  - **Updated continually for currency**
- **Applications registers condition triggers with the adaptation server**
  - **determine dynamic changes in sensor environment through the lookup servers or monitoring facilities**
  - **Activates the adaptation handler to select and execute the appropriate sensor fusion algorithm**
- **In heterogeneous sensor environment, the lookup server also cache info on the types of sensor devices and local detection algorithms**
  - **Sensors use this info to convert local detection info to a homogeneous data set for global data fusion**
  - **Accurate global fusion in spite of dynamic sensor changes and detection types mix**

Support for Adaptive Target Tracking

- Different algorithms for track fusion algorithms, e.g. linear Kalman Filter, weighted covariance fusion, Pheromone method, and Baynesian entity tracking
- Choice of algorithm may depend on the sensor and system characteristics, e.g. sensor signal to noise ratio, capability, etc.
  - Some computations are not beneficial under high noise conditions
  - Info may be cached in the lookup servers
- Example: If signal to noise ratio is low, may use simpler algorithm, e.g. simple fusion, but if it is high, use some computation and communication intensive algorithms , e.g. weighted covariance fusion
- Register triggers for these conditions with the adaptation server
- Info from lookup servers and monitoring facilities will trigger the adaptation handler to select and execute appropriate multi-sensor tracking algorithm
  - Could require less network bandwidth and computation time to get similar results

Conclusions

- **Distributed lookup, composition and adaptation services support self-organizing applications**
  - **New services and applications may be deployed spontaneously into existing sensor networks**
  - **Automatically respond to dynamically changing sensors and tasks, transparent to the application systems**
- **Provides simpler abstractions and communication model (e.g. remote exec) to applications systems**
- **With appropriate support in implementation of these distributed services on top of directed diffusion routing protocol, performance can be improved**